AURA

(Administrative Utilities for Resident Assistants)

An Honors Thesis (HONRS 499)

By

Wade A. Guisewhite

Thesis Advisor

Dr. Fu-Shing Sun

Ball State University

Muncie, Indiana

May 2009

Expected Date of Graduation:

May 9, 2009

Abstract

There are many facets of universities that most students do not experience in their four to five year tenure. I consider myself fortunate to have delved into a number of these, including the Department of Housing and Residence Life at Ball State University during my two-year employment as a Resident Assistant. While this position was a very positive experience for me, the volume of required paperwork was at times staggering.

In response to this, I set out to develop AURA (Administrative Utilities for Resident Assistants). AURA is a web application that facilitates the input of forms by resident assistants, and subsequent form retrieval by the appropriate housing personnel. This application will not only dramatically reduce the quantity of paper consumed by residence hall forms, but also serves to standardize the forms campus-wide.

Acknowledgements

I would like to express my gratitude and appreciation to the following persons, without whom this project would not have been possible:

- My advisor, Dr. Fu-Shing Sun, who has provided invaluable guidance and support throughout the project.
- Cindy Miller, Matt Kovach, Ro-Anne Royer Engle, Assistant Directors of Housing and Residence Life, and Dr. Cathy Bickel, Associate Director of Housing and Residence Life, all of whom have shared with me their time and insight. I extend a special thank you to Cindy Miller, with whom I worked most closely during this project's creation.
- University Computing Services, especially Phil Shaffer and Darren Terrell who met with me countless times and offered a multitude of technical advice.

AURA

(Administrative Utilities for Resident Assistants)

Wade Guisewhite

Table of Contents

1) Introduction

In the hierarchy of student employment at Ball State University, some positions are naturally more demanding than others. One such position is that of Resident Assistant. RAs take on a vital leadership role within residence halls, not only maintaining order and building community on each floor, but also taking on many administrative responsibilities that, at times, can carry a high overhead. Any RA can attest that the position's associated paperwork is cumbersome at best, and it is this dilemma that I have resolved to improve.

AURA is a web-based application that allows resident assistants to input required forms online rather than on paper. This results in the following immediate benefits, not only to the RA but also to residence hall directors, Housing directors, and the university as a whole:

- Reduced paper consumption in accordance with Ball State University's greening procedures
- Reduced clutter and need for physical organization mediums
- Standardization of forms across every residence hall on campus

While the concept of AURA is a simplistic one, its design and implementation proved to be much more challenging than I originally conceptualized. I proposed AURA to Dr. Cathy Bickel, Associate Director of Housing and Residence Life, in March of 2008. As my work on the project has very recently drawn to a close at the time of this writing, one year later, one can see that a significant amount of time and effort has gone into the application. This has been in part due to my desire for Ball State University to support and maintain AURA after my graduation. To effect this, I approached University Computing Services who, upon the approval of Dr. Alan Hargrave, Associate Vice President of Student Affairs and Director of Housing and Residence Life, agreed to assist me in this project.

Working with UCS has been a completely new experience for me, and while it carried a significantly steep learning curve, I have significantly benefited from the

partnership. The educational benefits I received will be discussed later in this document. I was assigned to work with Phil Shaffer, Database Administrator, and Darren Terrell, Senior Web Applications Developer. Although I have a significant amount of background experience both in the realm of database design as well as in the development of web applications, it became apparent very quickly that I was not quite up to speed on the current industry-level technology.

AURA was written using Microsoft ASP.NET as well as Microsoft C# programming languages. Data is stored using Microsoft SQL Server 2008, and all database queries are stored as procedures. Prior to working with UCS, I was unfamiliar with all of these technologies, and as such, was required to learn quite a large amount of information in a relatively short amount of time.

2) Specifications

While the purpose of AURA is simple, much thought and deliberation went into its actual specifications. For instance, one of the forms discussed for incorporation into AURA was the duty log. A duty log is a nightly report completed by each RA designated as "on duty" in each residence hall. This log contains information about happenings in the building, and any situations the RA may have attended to. Up until this point, the questions on the duty log were up to the discretion of each hall director. As AURA calls for standardization of forms, a uniform set of questions had to be decided upon.

To complete this task, I sent an e-mail to every hall director and assistant hall director asking for a copy of their duty log. I then compiled all of these questions into a master list, which I presented to the Housing directors. During this meeting, the directors and I discussed and ultimately decided upon the questions that would be included in the final product (initially, at least).

I also requested other forms from the hall directors as well, including weekly reports, program requests and evaluations, time away requests, and fund requests. Questions compiled from these forms were also discussed and pruned during the aforementioned meeting with the Housing directors, although they were not included in my final product due to time constraints.

The given specifications for Duty Log Entry (Role: RAs) are as follows:

- Information submitted with form not entered by user:
  This information will be displayed at the top of the form entry (generated by application.)
    - RA Name
    - Date / Time
- Information submitted with form entered by user:

  Date of Duty Shift:
  - Date Picker

Rounds:

Rounds will be displayed as a grid. When the page loads, there will be 3 round entries (time of round and applicable comments) displayed. If the user wishes to add additional rounds, a link below the rounds will permit this. The user is only required to submit 1 "round". Time of round will be a time picker.

Text:	Round x Time	
Input Type:	**Required?**	**Constraints:**
Time Picker	Y	Valid Time

Text:	Round x Comments	
Input Type:	**Required?**	**Constraints:**
Textbox	N	

Notes:

"Notes" are simply pieces of information which must be entered by the user. They are as follows:

Text:	Director on Duty	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Director on Duty Contact Number	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	What happened in the building tonight (programs, events, catastrophes, etc.)?	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Tell us about an interaction you had with a resident while on duty.	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Were there any security issues or policy violations? Did you submit an incident report?	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Were there any facilities or custodial problems with the building? Did you submit a TMA? If so, what is the TMA number?	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	TMA Number:	
Input Type:	**Required?**	**Constraints:**
Textbox	N	Integer

Text:	Were there any discrepancies with the fire equipment/doors/alarms?	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Do you have any problems/concerns/information we need to know about? Do we need to meet about this?	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Tell us something good that happened to you today!	
Input Type:	**Required?**	**Constraints:**
Textbox	Y	

Text:	Do you have any additional comments?	
Input Type:	**Required?**	**Constraints:**
Textbox	N	

The given specifications for Duty Log Retrieval (Role: Housing/Hall Directors) are as follows:

Text:	RA Name	
Input Type:	**Required?**	**Constraints:**
Textbox	N	

Text:	Start Date Range	
Input Type:	**Required?**	**Constraints:**
Date Picker	Y	

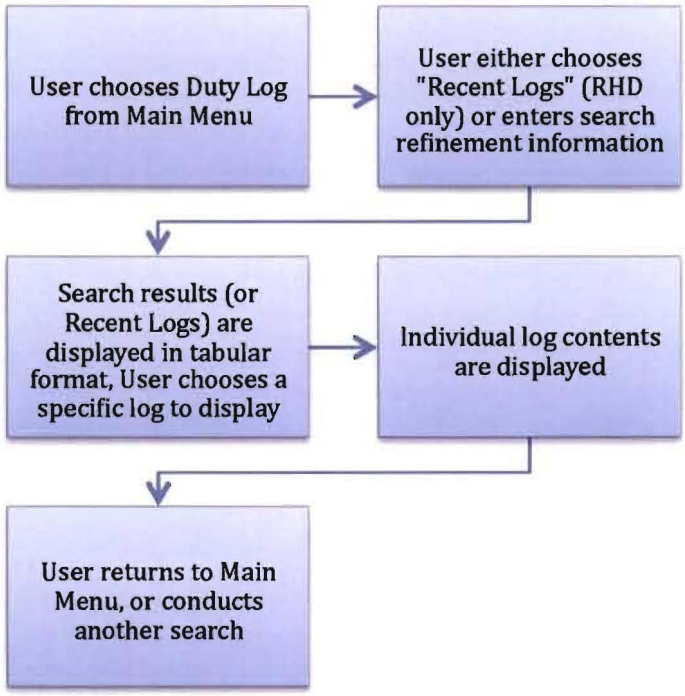
Text:	End Date Range	
Input Type:	**Required?**	**Constraints:**
Date Picker	Y	

Text:	Search Term	
Input Type:	**Required?**	**Constraints:**
Textbox	N	

Additionally, if the user is a Housing Director, there is the additional search field:

Text:	Residence Hall	
Input Type:	**Required?**	**Constraints:**
DropDown	Y	

The following is a flow diagram of the Duty Log Search process:



3) Implementation

3a) Presentation Tier

The presentation tier (the portion of the application the user actually sees and interacts with, e.g. page structure, layout, design, color, images, etc.) is constructed using the ASP.NET 3.5 framework. While I was initially intimidated by .NET, I have come to enjoy its extensive feature set. One of the principal improvements in ASP.NET versus the older, now-legacy Classic ASP is that individual controls on the page can be directly manipulated via the C# code file corresponding that page. For instance, when a button is clicked, a routine in the code file can be set to automatically run. This vastly improves a programmer's life as it makes the task of joining the presentation tier and the logic tier much easier.

The page layouts – including header, footer, color scheme, etc – are all part of UCS's pre-existing framework. This made my task easier in designing the user interface, as I needed only to develop the portions of the presentation tier directly manipulated by my application.

Additionally, UCS uses a suite of web controls developed by a company named Telerik. These handy and innovative tools include built-in date/time pickers, amongst many other things. The date/time picker is an icon that when clicked displays either a clock or a calendar from which the user may select the date/time without needing to bother with keyboard input.

The layout of each page is very similar, an intended design upon the part of UCS, to create a uniform style across all web applications. There is a navigation pane on the left side of the page, and any page content is shown on the right. Included in this report are screenshots of each page within the application, which may help to aid the reader in its visualization.

3b) Logic Tier

The logic tier (the actual code which determines the application's logical progression of events) is written in Microsoft C#. While I had never developed an application using C# prior to AURA, the learning curve was not substantial. C# is very similar in structure to Java, a well-known and very popular programming language designed by Sun Microsystems.

Certain components of AURA's logic were pre-built by UCS including authentication (logging into the application). This is of benefit to the end-user as it allows the use of existing BSU usernames and passwords, rather than needing to create a separate account solely for this application. And while a few other classes pertaining to fluidity within the UCS framework were provided to me, a great majority of the application's code was newly created.

The logic tier of AURA contains C# code files for each of the application's pages (i.e. DutyLogEntry, HallDutyLogSearch, HousingDutyLogSearch, etc.) These code files are responsible for error checking, sending input to data objects (which are in turn sent to the database), and any number of other logic-oriented tasks the application must carry out.

3c) Data/Object Tiers

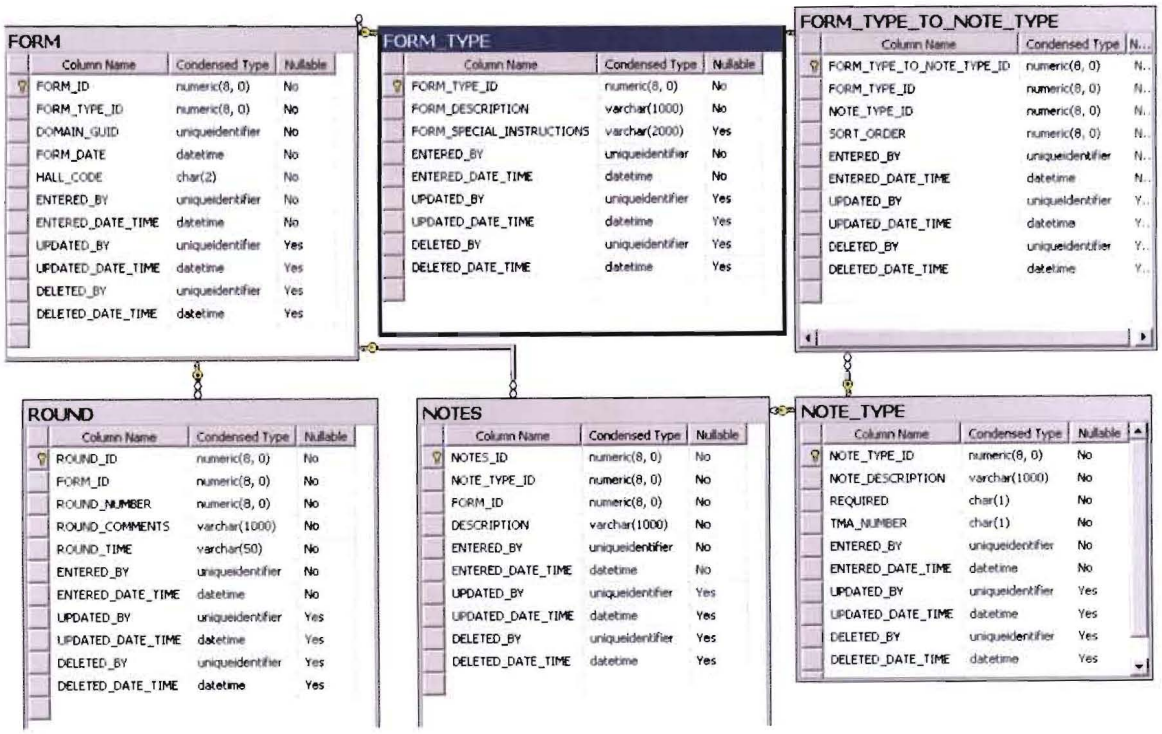
The Data and Object tiers work hand in hand to handle data structures and object definitions. Basically, for each type of data which may be needed within the application which is not part of the native C# codebase (e.g. data types explicitely defined by this application) a corresponding class must be created in the data tier and in the object tier.

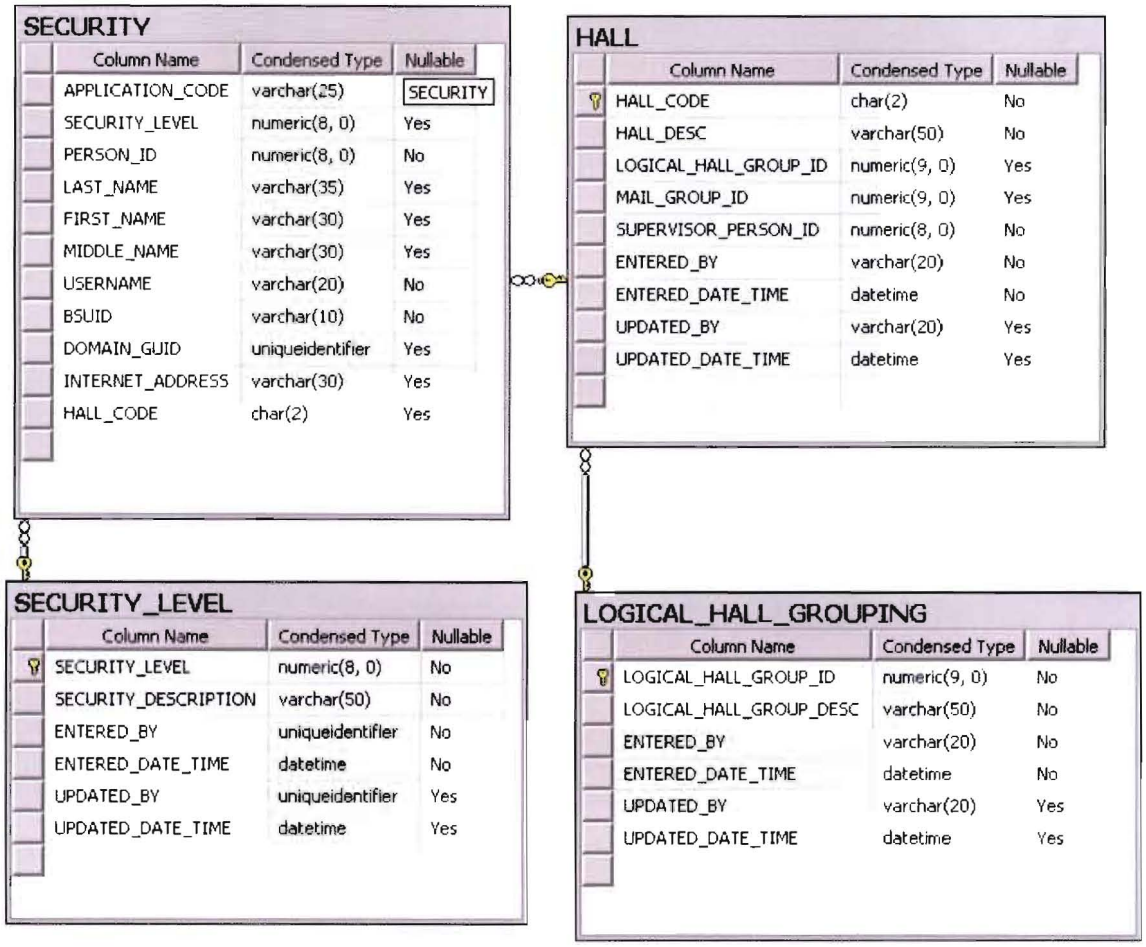
The data tier is primarily responsible for communicating with the database. When information is needed from the database or information must be committed to the database, the data tier is invoked. The distinction between the logic tier and data tier is a small, yet very important one. In past experiences, it never occurred to me that separating the two could be beneficial. However, in doing so, I have found that the organizational benefits are incredible. For instance, all database interactions for a specific type of data are all contained in one file, rather than spread out between each of the logic files. This cuts down on time spent searching through one's own code for needed information.

The level of technical knowledge required to fully understand the nuances of he data/object tier is beyond the scope of this report.

3d) Database

The database (some would consider this to be another tier of the application) is a completely separate entity from the application code, and is used to store all of the data which is generated or referenced by the application. Information is stored in tables, and each table groups together records of a certain sort. For instance, AURA's database contains a table to hold forms, a table to hold rounds, a table for notes, etc. The following is a graphical depiction of AURA's tables, and the relationships between them:





The following is a brief overview of each table:

- FORM – Contains one row for each form submitted
- FORM_TYPE – Contains the type of each form (i.e. Duty Log)
- FORM_TYPE_TO_NOTE_TYPE – Connects each note type to its corresponding form type (i.e. Which questions go with which form)
- ROUND – Contains one row for each round (If a user submits a form with 3 rounds, 3 entries are made to the ROUND table)
- NOTE – Contains one row for each note (question) submitted within a form
- NOTE_TYPE – Contains the note types, or questions, on each form
- SECURITY, HALL, SECURITY_LEVEL, LOGICAL_HALL_GROUPING – These four tables contain information identifying users and to which hall they belong

4) Meeting Log

During the year long process of creating AURA, I held many meetings with my advisor, with UCS, and with the directors of Housing and Residence Life. The following is a log of these meetings:

Date	Topic	With Whom
3/8/2008	Project Proposal	Cathy Bickel
3/21/2008	Initial Discussion	Cindy Miller
4/10/2008	Project Proposal	Fu-Shing Sun
4/18/2008	Project Proposal	Joanne Edmonds
5/1/2008	Project Proposal	Brian Means, UCS
8/29/2008	Initial Discussion	Brian Means, UCS
9/5/2008	Initial DB Design	Phil Shaffer
9/11/2008	Requirements	Fu-Shing Sun
10/2/2008	DB Design	Phil Shaffer
10/14/2008	DB/Procedures	Phil Shaffer / Darren Terrell
10/16/2008	DB/Procedures	Phil Shaffer / Darren Terrell
10/21/2008	Form Compilation	Housing Directors
10/28/2008	DB/Procedures	Phil Shaffer / Darren Terrell
11/4/2008	DB/Procedures	Phil Shaffer
11/11/2008	DB / Data Layer	Phil Shaffer / Darren Terrell
11/13/2008	DB / Data Layer	Phil Shaffer / Darren Terrell
11/17/2008	Progress Report	Fu-Shing Sun
11/18/2008	DB / Data Layer	Phil Shaffer / Darren Terrell
11/25/2008	DB / Data Layer	Phil Shaffer / Darren Terrell
12/5/2008	Progress Report	Fu-Shing Sun
12/9/2008	Data Layer	Darren Terrell
12/12/2008	Project Questions	Dean Ruebel
1/23/2009	Progress Report	Fu-Shing Sun
1/28/2009	Milestone Discussion	Phil Shaffer / Darren Terrell
1/30/2009	Progress Report	Fu-Shing Sun
2/5/2009	Data / Logic Tiers	Darren Terrell
2/6/2009	Progress Report	Fu-Shing Sun
2/10/2009	Data / Logic Tiers	Darren Terrell
2/19/2009	Logic Tier	Darren Terrell
2/20/2009	Progress Report	Fu-Shing Sun
3/3/2009	Logic Tier	Darren Terrell
3/6/2009	Progress Report	Fu-Shing Sun
3/20/2009	Presentation Tier	Darren Terrell
4/7/2009	Presentation Tier	Darren Terrell
4/16/2009	Presentation Tier	Darren Terrell
4/28/2009	Finalization	Darren Terrell
4/29/2009	Acceptance	Housing Directors
5/6/2009	Finalization	Darren Terrell
5/6/2009	Acceptance / Grading	Fu-Shing Sun

5) Conclusions

5a) Educational Impact

Upon undertaking this project, I expected to gain additional experience in a field with which I was already very familiar, but was not expecting to pick up a great number of new skills. As aforementioned, due to working with University Computing Services, this initial analysis was disproven. In short, I have learned an extraordinary number of things throughout the completion of this project. I have learned not only an entirely new programming language, but also a set of industry standards by which to format and maintain application code. I have learned how to use Microsoft SQL Server, and how to create and properly format stored procedures. Additionally, I have learned about source code version control, and the check-in/check-out process.

However, the most important learning curve of this project stemmed from working with Darren Terrell and Phil Shaffer of UCS. In previous programming endeavors, without any professional guidelines or accountability, I often found myself content with "whatever works". This modus operandi does not sit well with UCS, however, and Darren and Phil were quick to break me of the habit. They held me accountable for my work, and challenged me to develop my application to a higher standard. Many of the coding practices and nuances that I learned from them I will undoubtedly carry into my professional career.

5b) What Has Been Accomplished

AURA is currently a completely functional web application which handles the input and search/retrieval of Duty Logs by Resident Assistants, Hall Directors, and Directors of Housing and Residence Life. Although my original goal to implement a variety of different forms into this application was not fulfilled, and there is still plenty of future work to be done, the Department of Housing and Residence Life has accepted the application. After it has undergone testing and has been declared

production-ready, AURA will be deployed on Ball State University's website for use by the Department of Housing and Residence Life.

5c) Future Considerations

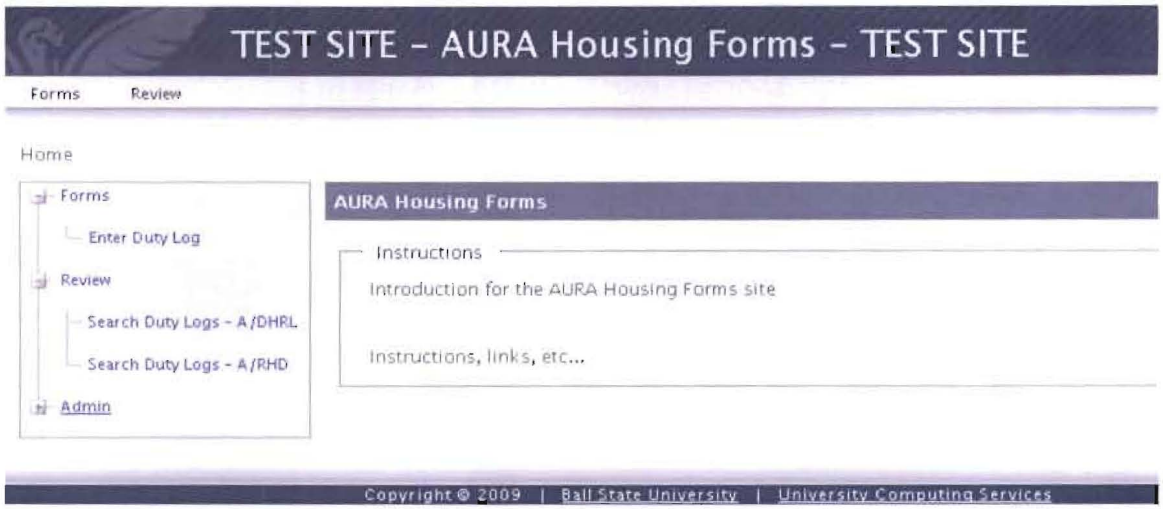
As with any completed work, there is always room for improvement and extension. AURA was designed with modularity and flexibility in mind, and as such, can be easily modified to handle additional forms. Should the Department of Housing and Residence Life find success with AURA and choose to widen its scale, they need only submit a request to UCS to complete the necessary modifications.

AURA is also equipped to handle modifications on a smaller scale, as well. As "note types" are stored in the database, should Housing decide to add additional questions to an existing form type, the only required modification is the addition of a row to the database.

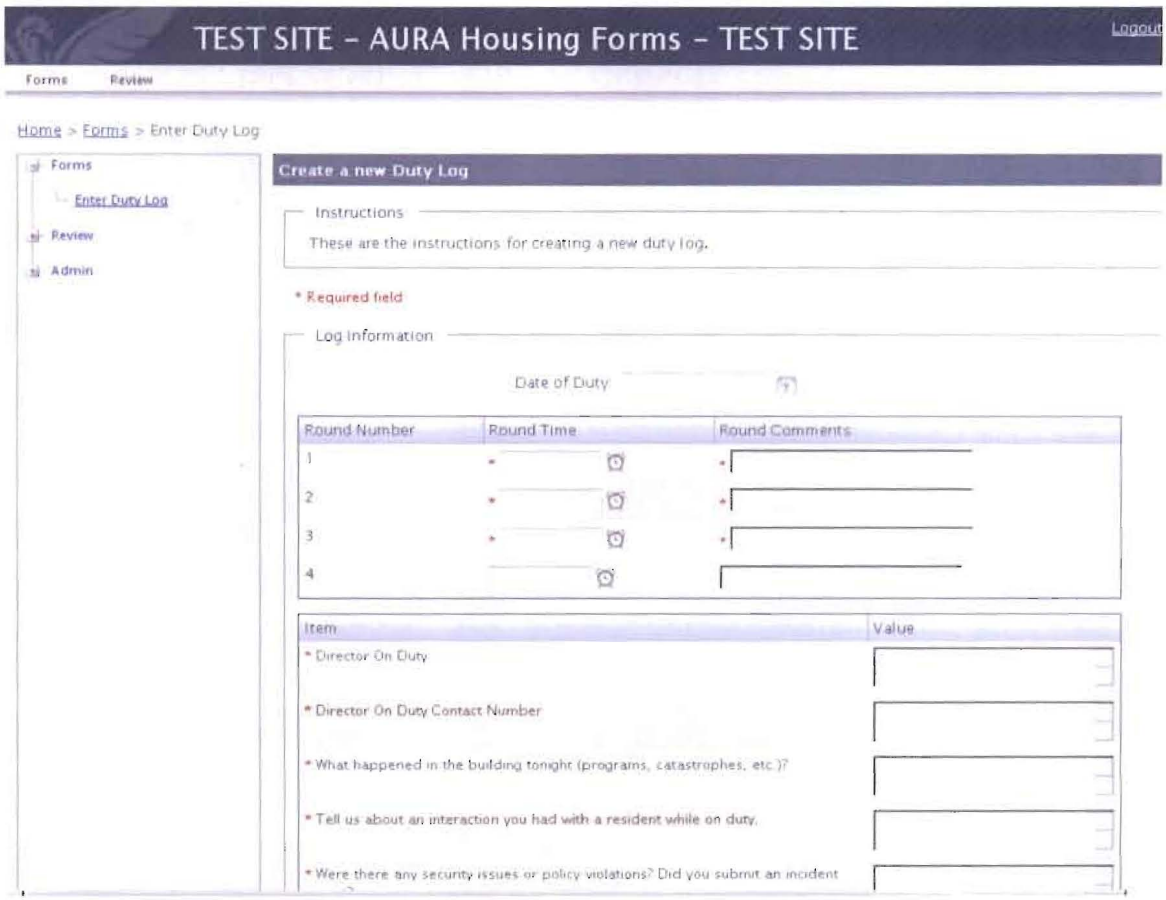
With a bit of time and effort, AURA could eventually be modified to make these changes from within the application itself, ultimately culminating in an application where new forms can be created on the fly. Were its evolution to reach this stage, AURA would be virtually self-sustaining as no modifications to the application code would be necessary to carry out these tasks.

6) Screenshots

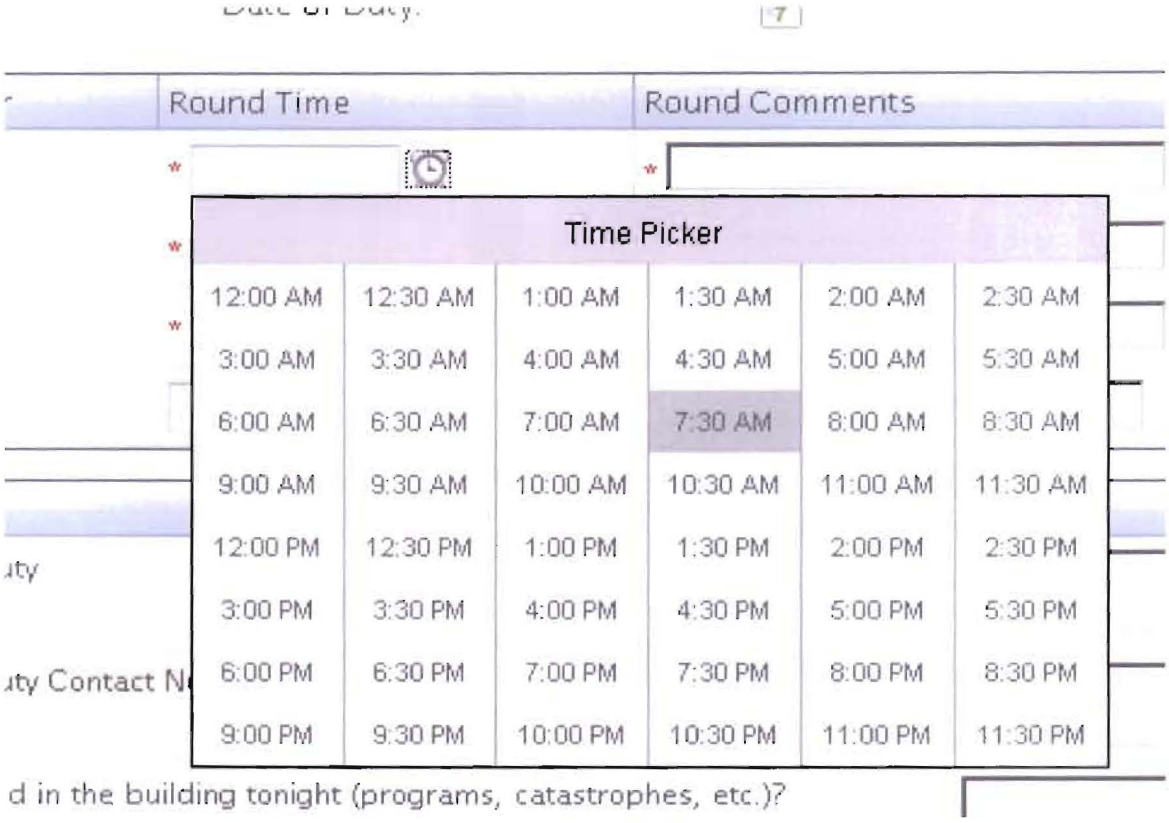
- Main application page



- Duty Log entry form

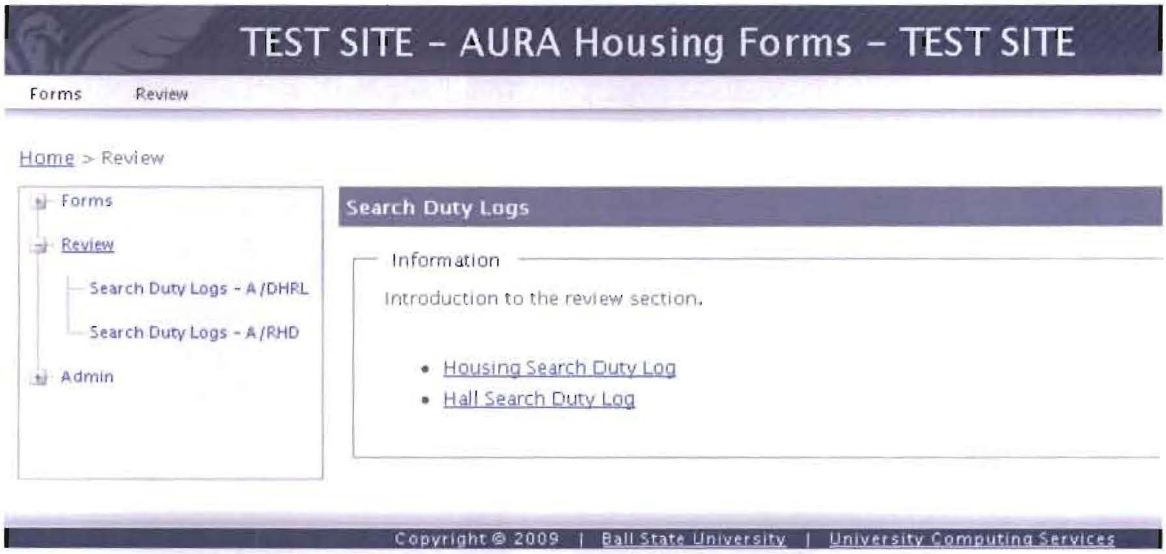


- Duty Log entry form time picker



- Duty Log entry form date picker

- Hall/Housing Director Main Screen



- Search Criteria

- Search Results

Search Results

Date	RA Name
4/29/2009	Bickel, Catherine

- Individual Log Displayed